

Introduction to Linkspers BOS

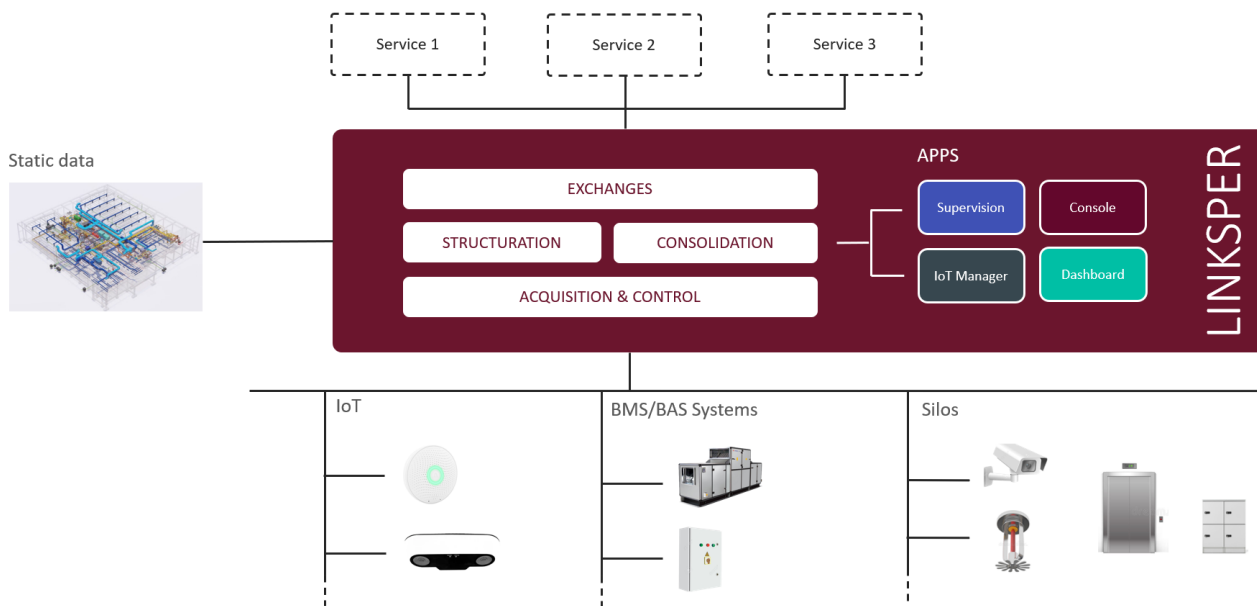
Linkspers is a **Building Operating System**: a software platform integrating data from heterogeneous sources into a single unified interface to be shared with third parties.

Linkspers Stack

Linkspers is made of 4 main pillars:

- **Acquisition & Control**: Linkspers communicates with almost any OT & IT systems such as traditional BMS/BAS systems, IoT systems (both local or cloud based) and silos
- **Structuration**: Linkspers ingests static data models such as BIM, CMMS hierarchies... and pair them with dynamic data to create contextualized data
- **Consolidation**: Linkspers provides tools to create local scenarios, generate syntheses, global commands...
- **Exchanges**: Linkspers provides a default API for third parties as well as many out of shelves connectors to major platforms or technologies (Azure, GCP, AWS, Elasticsearch, pure MQTT, MongoDB...)

All of this is managed through applications which help the BOS integrator to administer data, monitor the interactions and handle the evolutions.



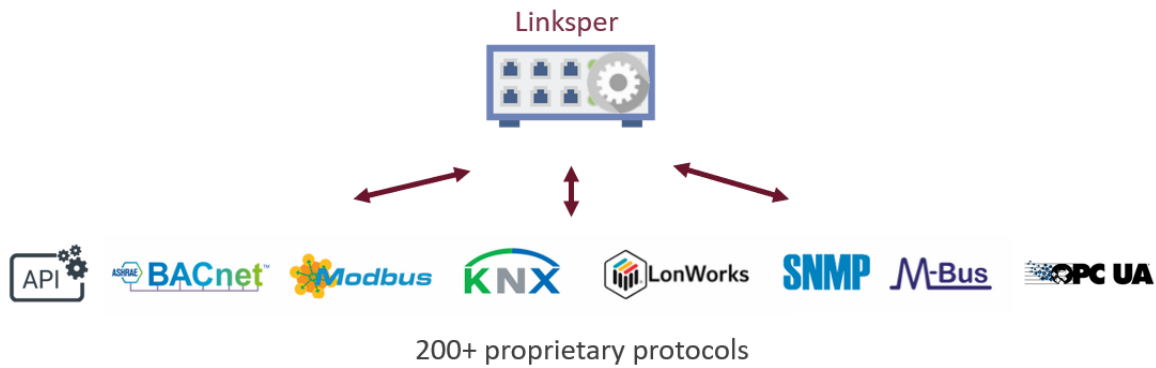
Acquisition & Control

Linkspers is based on the Niagara™ platform from Tridium which is fully agnostic from any hardware. A lot of communication drivers are available both for field and IP protocols: BACnet (Client or Server), Modbus (Master and Slave), LonWorks, KNX, MBus, SNMP, OPC UA, MQTT... More than 200 proprietary protocols are also available to retrieve data from existing and ancient installations (Cbus, N2, TAC, Sauter, Trend IQ...).

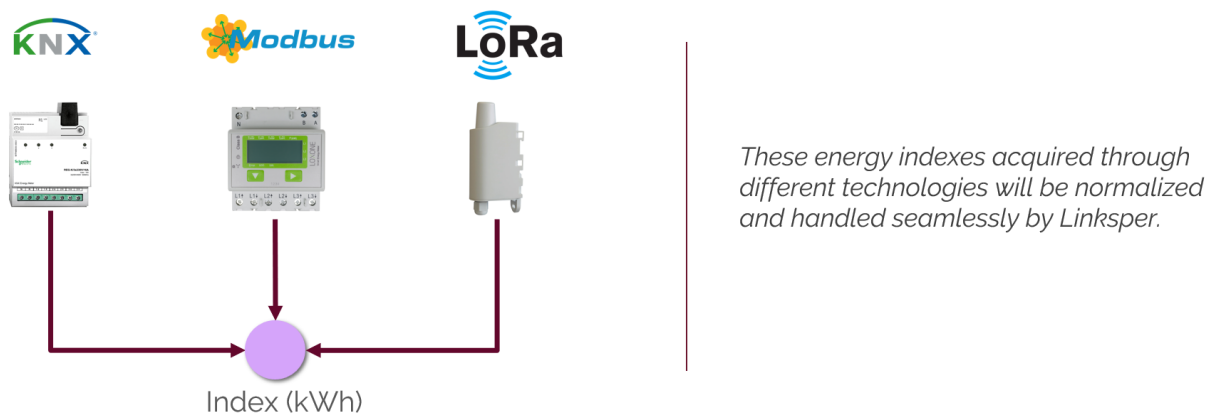
One of the key advantage of Linkspers is to have this capacity to connect as direct as possible to any equipment, avoiding additional gateways in the middle that would add complexity and weakness.

More and more systems (especially silos & IoT) are exposing their data through Rest API. Linkspers has a built-in generic driver for REST API allowing it to communicate with any system without any need for specific development, it is just a bit of configuration.

In case a new driver is necessary, Linkspers provides an advanced SDK (Software Development Kit) to create new drivers.



Linkspers normalizes data no matter its origin and its technology into a single and unified representation.



Any data acquired, regardless of the protocol, is thus transformed into the same format and allows the construction of logic and algorithms from a simple and intuitive GUI. This does not require any development.

- Creation of control logics (mathematical functions, boolean logics, conditional logics, control loop... without any need for development or scripts)
- Generation of alarms, indicators and summaries
- Integrated query tools (operating reports, debugging, filters, dynamic rankings)
- Event management
- Planning
- Data pre-processing (aggregation, accumulation and scalar functions over time, running time...)

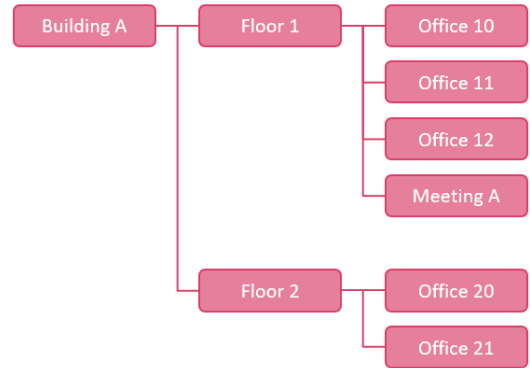
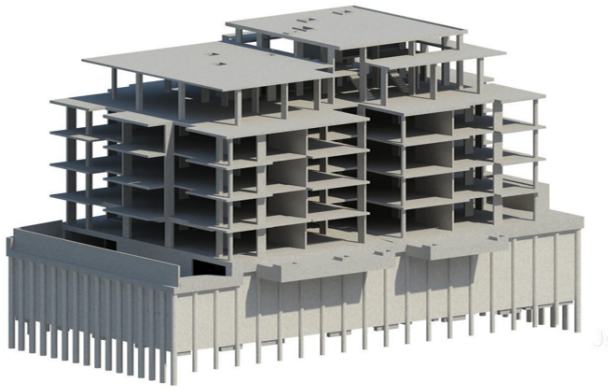
Learn more about [Data Types](#)

Structuration

Linkspers integrates a graph database used to provide a context to each acquired data making it easier to represent complex systems such as buildings and all their subsequent assets. Third party services can use this graph database to automatically (machine to machine) understand all the subsystems and assets involved.

Spatial data model

The first data model describes the building's structure as shown below:



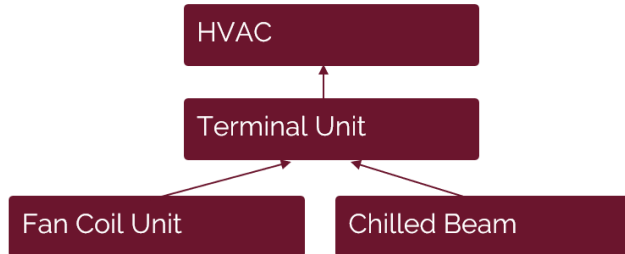
Linksper is implemented with an IFC-based model consisting of a main Site/Building/Floor/Space/Equipment tree and the ability to describe zones. The model is custom built for the site.

It can be originated from a BIM model or built from other sources. Linksper has a lot of tools to ingest a BIM model and handles its evolution across a building's evolution.

Other data models

Linksper integrates other data models such as an ontology of all equipment to describe the nature of each asset Linksper is acquiring data from. This data model is standardized and therefore uniform across all implemented Linksper BOS.

This allows any third party to understand data from a Linksper instance no matter the integrator, no matter the hardware installed on site.



Linksper integrates a data model to describe the nature of each data point to define whether it is a measurement, a state, its associated dimension, the resource it handles...



Learn more about [Data models](#)

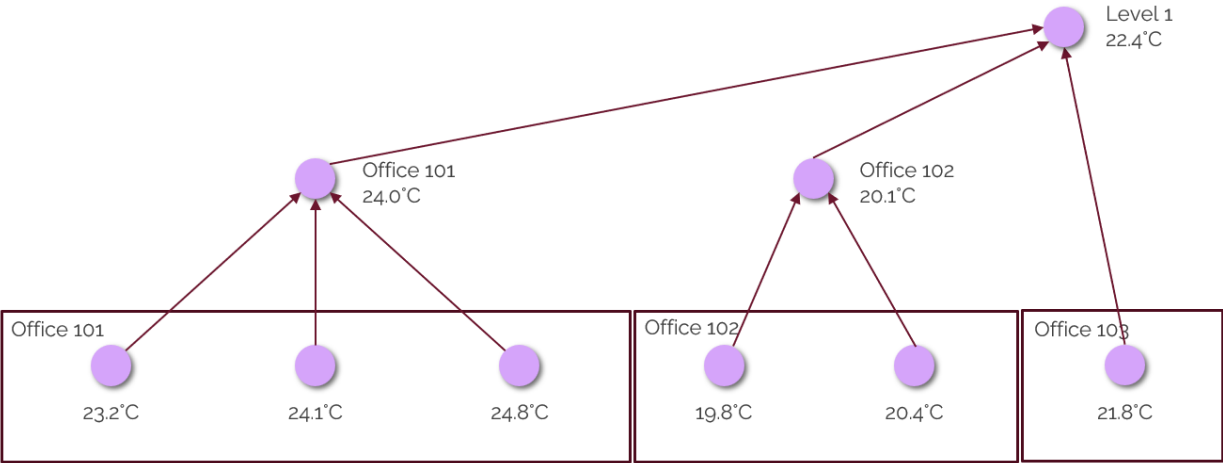
Consolidation

Syntheses

Once the data are structured with a context, Linkspers builds on the top consolidated data such as real-time syntheses. They are usually created from the spatial environment of the building and creates pyramidal data.

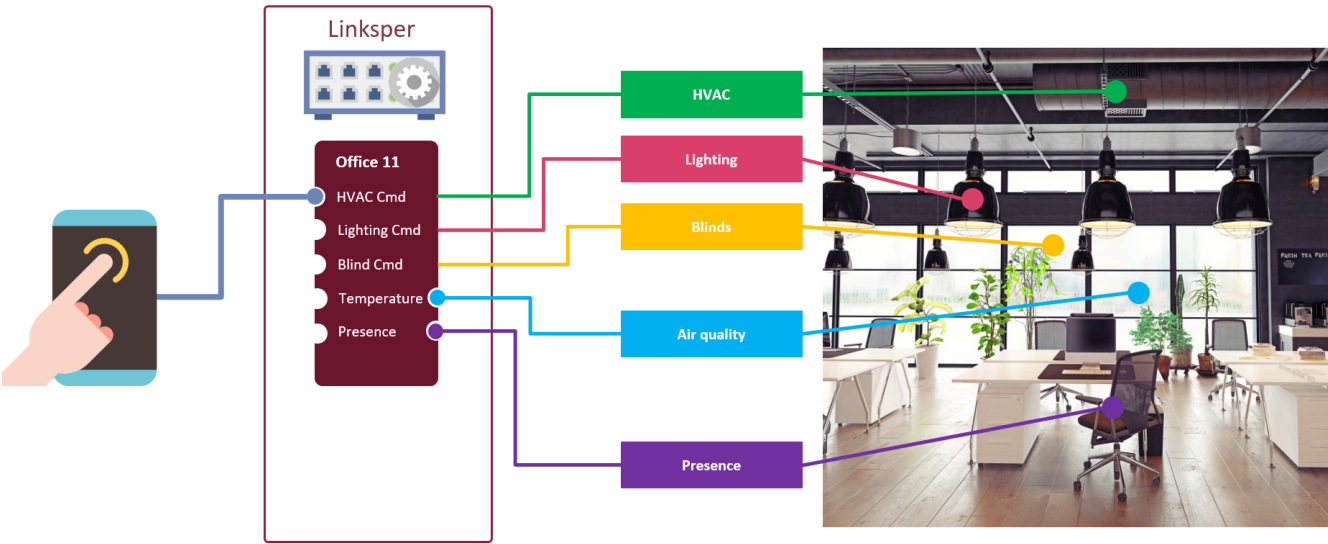
Example below where several temperature sensors are acquired per office, an average value is computed for each room as well as for the floor. A third party can then decide to ask the BOS for the original temperature sensor, the average per room or per floor.

A list of standardized syntheses is available.



Global commands

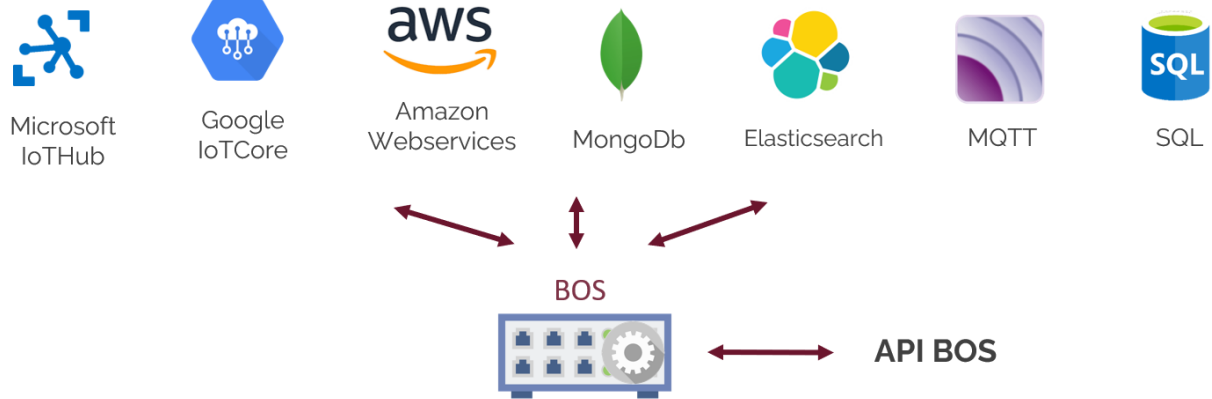
This is also available for commands. To avoid a third party to control each light of a room, it can send a global command to dim all the lights of the room. The BOS will handle the various number of lights on each room.



Exchanges

Data connectors

Linkspers integrates by default several data connectors to major cloud platforms as well to independent databases. Connectors can be used for real-time exchanges or timeseries export.



Linkspier API

Linkspier provides as well a RESTful API to allow a third party to communicate with it.

Authentication Authenticate to the API

POST `/oauth/token` Authenticate using jwt

Explore Explore the BOS hierarchies and their respective assets

GET `/v1/explore/hierarchies` Explore all hierarchies

GET `/v1/explore/hierarchies/{name}` Explore a single hierarchy

GET `/v1/explore/assets/{id}` Explore an asset

GET `/v1/explore/assets/{id}/alarms` Explore the alarms of the given asset

GET `/v1/explore/assets/{id}/devices` Explore the devices of the given asset

GET `/v1/explore/assets/{id}/equipment` Explore the equipment of the given asset

GET `/v1/explore/assets/{id}/histories` Explore the histories of the given hierarchy